Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A method of operating a processor comprising:

receiving data in a processing thread having a processing thread number; and

loading the data into selected bits of a register ~~having a register address corresponding~~ according to the processing thread number.

2. (Original) The method of claim 1 wherein the register is a control and status register (CSR).

3. (Currently amended) The method of claim 2 wherein the ~~status and~~ control and status register ~~in contained in~~ is coupled to a 64-bit wide first-in first-out (FIFO) bus.

4. (Original) The method of claim 3 wherein the FIFO bus interfaces with Media Access Controller (MAC) devices.

5. (Currently amended) The method of claim 1 wherein the data represents ~~values~~ hexadecimal mask values 0 to ~~hexadecimal mask~~ 0x3FF.

6. (Original) The method of claim 1 wherein the processing thread represents processing in a micro engine of a multi-threaded processor.

7. (Currently amended) The method of claim 1, wherein loading the data ~~shifting~~ comprises:

shifting a first portion of the data by an amount equal to the processing thread number; and

shifting a second portion of the data into bits corresponding to a breakpoint (BP) register 2 ~~to a~~ through a breakpoint BP register 0.

8. (Original) The method of claim 1 wherein receiving the data further comprises receiving a token.

9. (Original) The method of claim 8 wherein the token represents overriding qualifiers.

10. (Original) The method of claim 8 wherein the token is a 32-bit word.

11. (Currently amended) The method of claim 10 wherein a token format comprises:

an OV field in bit 31;

[[an]] a micro engine (UENG) ADDR field in bits 30:28;

a reserved field in bits 27:16;

an OV field in bit 15;

a fast write data field in bits 14:5;

a reserved field in bits 4:3;

an OV field in bit 2; and

a CTX field in bits 1:0.

12. (Original) The method of claim 11 wherein a micro engine address overrides a default micro engine address if bit 31 is set.

13. (Original) The method of claim 11 wherein bits 30:28 specify a micro engine associated with a control and status register (CSR).

14. (Original) The method of claim 11 wherein bits 27:16 return 0 when read.

15. (Original) The method of claim 11 wherein a micro engine address overrides a default micro engine address if bit 15 is set.

16. (Original) The method of claim 11 wherein bits 14:5 represent valid data to be written to a control and status register (CSR).

17. (Original) The method of claim 11 wherein bits 4:3 return 0 when read.

18. (Original) The method of claim 11 wherein a context (CTX) field overrides a default context if bit 2 is set.

19. (Original) The method of claim 11 wherein bits 1:0 specify a context associated with a control and status register (CSR) reference.

20. (Currently amended) A computer program product, disposed on a computer readable medium, the program comprising instructions for causing a computer to:

receive data in a processing thread having a processing thread number; and

load the data into selected bits of a register ~~corresponding~~ according to the processing thread number.

21. (Original) The computer program product of claim 20 wherein the register field is a control and status register (CSR).

22. (Currently amended) The computer program product of claim 21 wherein the ~~status and~~ control and status register is ~~contained in~~ coupled to a 64-bit wide first-in first-out (FIFO) bus.

23. (Original) The computer program product of claim 22 wherein the FIFO bus interfaces with Media Access Controller (MAC) devices.

24. (Currently amended) The computer program product of claim 20 wherein the data represents ~~values~~ hexadecimal mask values 0 to ~~hexadecimal mask~~ 0x3FF.

25. (Original) The computer program product of claim 20 wherein the processing thread represents processing in a micro engine of a multi-threaded processor.

26. (Original) The computer program product of claim 20 further comprising instructions for causing the computer to:

shift a first portion of the data left by an amount equal to the processing thread number; and

shift a second portion of the data into bits corresponding to a breakpoint (BP) register 2 ~~to a~~ through BP register 0.

27. (Original) The computer program product of claim 20 further comprising an instruction for causing the computer to receive a token.

28. (Original) The computer program product of claim 27 wherein the token represents overriding qualifiers.

29. (Original) The computer program product of claim 27 wherein the token is a 32-bit word.

30. (Currently amended) The computer program product of claim 29 wherein a token format comprises:

an OV field in bit 31;

[[an]] a micro engine (UENG) ADDR field in bits 30:28;

a reserved field in bits 27:16;

an OV field in bit 15;

a fast write data field in bits 14:5;

a reserved field in bits 4:3;

an OV field in bit 2; and

a CTX field in bits 1:0.

31. (Original) The computer program product of claim 30 wherein a micro engine address overrides a default micro engine address if bit 31 is set.

32. (Original) The computer program product of claim 30 wherein bits 30:28 specify a micro engine associated with a control and status register (CSR).

33. (Original) The computer program product of claim 30 wherein bits 27:16 return 0 when read.

34. (Original) The computer program product of claim 30 wherein a micro engine address overrides a default micro engine address if bit 15 is set.

35. (Original) The computer program product of claim 30 wherein bits 14:5 represent valid data to be written to a control and status register (CSR).

36. (Original) The computer program product of claim 30 wherein bits 4:3 return 0 when read.

37. (Currently amended) The computer program product of claim 30, wherein a context (CTX) field overrides a default context if bit 2 is set.

38. (Currently amended) The computer program product of claim 30 wherein bits 1:0 specify a context associated with a control and status register (CSR) reference.~~ABSTRACT~~

~~A method of operating a processor including receiving data in a processing thread having a processing thread number and shifting the data into a register corresponding to the processing thread number.~~